TIKTOK VIDEO OUTCOME - WATCH THESE VIDEOS FIRST (linked)

Instruction:

Setting Up the Script

1. Copy the Script below: Depends on the need.
2. Create a New Google Apps Script:
    a. Navigate to Google Drive.
    b. Click on “New” > “More” > “Google Apps Script.”
    c. Delete any pre-existing code and paste the provided script into the code editor.
3. Authorize the Script:
    a. Click on the disk icon or navigate to "File" > "Save" and name your project.
    b. Click on the run (play) icon in the toolbar, and select the main function to execute.
    c. If it's your first time running the script, you will be prompted to review permissions. Click on "Review Permissions," choose your account, and click "Allow."

Version 1: Sort Files (Based on File Types)

```
function organizeMyFiles() {
 var driveService = DriveApp;
 var masterFolderIterator = driveService.getFoldersByName("Sorted Files");
 var masterFolder;
  if (masterFolderIterator.hasNext()) {
   masterFolder = masterFolderIterator.next();
 } else {
   masterFolder = driveService.createFolder("Sorted Files");
 }
  var rootFolder = driveService.getRootFolder();
 var allFiles = rootFolder.getFiles();
  while (allFiles.hasNext()) {
   var individualFile = allFiles.next();
   var mimeType = individualFile.getMimeType();
```

```
    if (mimeType === MimeType.GOOGLE_APPS_SCRIPT) {
      continue;
    }

    var category = categorizeFile(individualFile);
    var targetFolder = fetchOrCreateFolder(masterFolder, category);
    individualFile.moveTo(targetFolder);
  }
}

function categorizeFile(fileItem) {
  var fileType = fileItem.getMimeType();
   switch (fileType) {
    case MimeType.GOOGLE_DOCS:
      return "Documents";
    case MimeType.GOOGLE_SHEETS:
      return "Spreadsheets";
    case MimeType.GOOGLE_SLIDES:
      return "Presentations";
    case MimeType.GOOGLE_FORMS:
      return "Forms";
    case MimeType.JPEG:
    case MimeType.PNG:
    case MimeType.GIF:
      return "Pictures";
    case MimeType.PDF:
      return "PDF";
    case MimeType.MP4:
    case MimeType.AVI:
    case MimeType.MPEG:
    case MimeType.QUICKTIME:
    case MimeType.MOV:
```

```
      return "Video";
    default:
      return "Other";
  }
}

function fetchOrCreateFolder(mainFolder, folderLabel) {
  var subFolderIterator = mainFolder.getFoldersByName(folderLabel);
   if (subFolderIterator.hasNext()) {
    return subFolderIterator.next();
  } else {
    return mainFolder.createFolder(folderLabel);
  }
}
```

Version 2: Sort Files inside Folders (By Year)

```
function organizeAllFilesByYear() {
  var driveService = DriveApp;
  var masterFolderIterator = driveService.getFoldersByName("Sorted Files by Year");
  var masterFolder;
   if (masterFolderIterator.hasNext()) {
    masterFolder = masterFolderIterator.next();
  } else {
    masterFolder = driveService.createFolder("Sorted Files by Year");
  }
   var allFiles = driveService.getFiles();
   while (allFiles.hasNext()) {
    var individualFile = allFiles.next();
    var dateCreated = new Date(individualFile.getDateCreated());
    var yearCreated = dateCreated.getFullYear();
```

```
   var targetFolder = fetchOrCreateYearFolder(masterFolder, yearCreated);
   individualFile.moveTo(targetFolder);
 }
}

function fetchOrCreateYearFolder(mainFolder, year) {
 var folderName = year <= 2016 ? '2016 and Older' : year.toString();
 var subFolderIterator = mainFolder.getFoldersByName(folderName);
  if (subFolderIterator.hasNext()) {
   return subFolderIterator.next();
 } else {
   return mainFolder.createFolder(folderName);
 }
}
```

Version 3: Sort Files Outside Folders (By Year)

```
function organizeMyFilesByYear() {
 var driveService = DriveApp;
 var masterFolderIterator = driveService.getFoldersByName("Sorted Files by
Year");
 var masterFolder;
  if (masterFolderIterator.hasNext()) {
   masterFolder = masterFolderIterator.next();
 } else {
   masterFolder = driveService.createFolder("Sorted Files by Year");
 }
  var rootFolder = driveService.getRootFolder();
 var allFiles = rootFolder.getFiles();
  while (allFiles.hasNext()) {
```

```
   var individualFile = allFiles.next();
   var dateCreated = new Date(individualFile.getDateCreated());
   var yearCreated = dateCreated.getFullYear();
   var targetFolder = fetchOrCreateYearFolder(masterFolder, yearCreated);
   individualFile.moveTo(targetFolder);
 }
}

function fetchOrCreateYearFolder(mainFolder, year) {
 var folderName = year <= 2016 ? '2016 and Older' : year.toString();
 var subFolderIterator = mainFolder.getFoldersByName(folderName);
  if (subFolderIterator.hasNext()) {
   return subFolderIterator.next();
 } else {
   return mainFolder.createFolder(folderName);
 }
}
```